Incremental Few-Shot Learning with Attention Attractor Networks

Mengye Ren, Renjie Liao, Ethan Fetaya, Richard S. Zemel
University of Toronto and Vector Institute

• Testing **on only new classes** in “few-shot” is not natural.
• **Incremental few-shot learning**: learn new classes on top of old classes. **No access to the old data.**
• At each test episode, learn a linear classifier **until convergence**.
• Attention over base classes to form **attractor regularizers**.
• At the end of the episode, test on a **query set of both base and novel**.
• Use **recurrent backprop (RBP)** instead of **truncated BPTT** for learning more **stable loss functions**.
• **Learned regularizers** significantly reduce class interference.
Auto-Meta: Automated Gradient Based Meta Learner Search

Jaehong Kim¹, Sangyeul Lee¹, Sungwan Kim¹, Moonsu Cha¹, Jung Kwon Lee¹, Youngduck Choi¹,², Yongseok Choi¹, Dong-Yeon Cho¹, and Jiwon Kim¹

¹ SK T-Brain  ² Yale University

Automated architecture search

Gradient-based Meta learning

Performance improvement

Few-shot image classification

(Omniglot, Mini-ImageNet)
We propose a framework for meta-learning across task geometries by learning from gradient trajectories.

We present Leap, a light-weight meta-learner that scales beyond few-shot learning to tasks requiring millions of gradient steps.
Few-shot Learning For Free by Modelling Global Class Structure

Xuechen Li*, Will Grathwohl*, Eleni Triantafillou*, David Duvenaud, Richard Zemel

- Most approaches to few-shot classification use **episodic training**.
- We advocate for a simpler approach: a generative model over all classes: a VAE with a **mixture of Gaussians prior**.
- Few-shot learning is done by **variational inference**.
- Our model solves 3 tasks:
  - Few-shot classification
  - Few-shot generation
  - More realistic: **Few-shot integration**.
- Omniglot experiments:
  - On par with state-of-the-art on few-shot classification.
  - Largely outperform our baseline on few-shot integration.
TAEML: Task-Adaptive Ensemble of Meta-Learners

Workshop on Meta-Learning (MetaLearn2018)

Fig1. Current meta-learning for few-shot classification

Fig2. Solving to few-shot classify the birds: Training all of the tasks won’t be efficient

Fig3. Target task adaptive ensemble of pre-trained meta-learners
A Simple Transfer-Learning Extension of Hyperband
Lazar Valkov, Rodolphe Jenatton, Fela Winkelmolen, Cédric Archambeau

- Setting: Hyperparameter Optimisation
- Hyperband (HB):
  - Incrementally allocates more resources to the best-performing candidates initially taken from a pool of randomly sampled candidates.
  - Evaluates different number of initial candidates $n_i$ for $r_i$
- We enhance HB with model-based sampling, using ABLR (Peronne et al.)

$$P(y_t | w_t, z, \beta_t) = \mathcal{N}(\Phi_z(X_t, r_i)w_t, \beta_t^{-1}I_{N_t})$$

- Benefits:
  - Makes use of all data produced by a HB run
  - Can use data from past HB runs to learn better basis function
  - We don’t use heuristics for low number of data points, nor to encourage exploration
Learned optimizers that outperform SGD on wallclock and test loss

Luke Metz, Niru Maheswaranathan, Jeremy Nixon, C. Daniel Freeman, Jascha Sohl-Dickstein

Existing optimizers are hand designed. Can we do better with learning?

One popular strategy for training such optimizers is to leverage gradients and truncated backpropagation through time.

These methods, however, are notoriously unstable!

Careful choice of step length is required:
- Long truncations: exploding gradients
- Short truncations: biased gradients

We use variational optimization to "smooth" the loss surface by convolving it with a Gaussian.

\[ \mathcal{L}(\theta) = \mathbb{E}_{\tilde{\theta} \sim \mathcal{N}(\theta, \sigma^2 I)} \left[ L(\tilde{\theta}) \right] \]

To optimize this objective, we combine multiple gradient estimators with difference variances.

We train simple MLP-based learned optimizers that are faster in wallclock time and generalize better than existing hand-designed methods.
Learning to Learn with Conditional Class Dependencies

Xiang Jiang$^{1,2}$, Mohammad Havaei$^1$, Farshid Varno$^{1,2}$, Gabriel Chartrand$^1$, Nicolas Chapados$^1$, Stan Matwin$^2$

$^1$ Imagia Inc.  $^2$ Dalhousie University

Integrates two views of the data
The metric space captures class dependencies
Conditional batchnorm helps class separation
Unsupervised learning is commonly used as pre-training for downstream learning.

- We improve upon this by incorporating knowledge about the downstream task type: image classification.

**Unsupervised meta-learning** via CACTUs: meta-learning over tasks constructed from unlabeled data.

1. run embedding learning
   \[ \{x_i\} \xrightarrow{\text{embedding function}} \{z_i\} \]

2a. cluster embeddings multiple times
   \[ P_1, P_2, \ldots \]

2b. automatically construct tasks without supervision
   \[ \{x\} \xrightarrow{\text{train-test}} \{z\} \xrightarrow{\text{meta-learner } \mathcal{M}} \text{learning procedure } \mathcal{F} \]

3. run meta-learning on tasks
   \[ T_1, T_2, \ldots \]

Results: better than unsupervised learning, worse than supervised meta-learning
We develop a Bayesian meta-learning model that is capable of fast, efficient online updates and is trained for multi-step probabilistic predictions.

Using this model, we build a control algorithm that captures online model uncertainty and automatically trades off safety and performance.
Learning to Adapt in Dynamic, Real-World Environments Through Meta-Reinforcement Learning

Anusha Nagabandi*, Ignasi Clavera*, Simin Liu, Ron S. Fearing, Pieter Abbeel, Sergey Levine, Chelsea Finn

Goal

Use recent experiences to quickly adapt to the current situation.

Train time: Learning to Adapt

Meta-learn a dynamics model

Tasks: temporal windows

Objective:

\[
\min \mathcal{L}(D^r_t, \theta_t) \quad \text{s.t.} \quad \theta_t = \mathcal{M}_\psi(D^r_t, \theta) 
\]

\(D^r_t \rightarrow \) Future data

\(D^r_t \rightarrow \) Past data

Test time: Meta-Model-Based RL

Meta-trained prior \(\theta^*\)

Update rule \(u_\psi\)

Adapted model \(\theta^{**}\)

MPC controller (MPPI)

Experiments

Pier  Terrain slopes  Disabled  Crippled  Slope  Pose error  Payload  Missing leg
Learning to Design RNA

Frederic Runge* Danny Stoll* Stefan Falkner Frank Hutter

- **Meta-learn** a policy across RNA Design tasks
- **AutoML** for joint optimization of:
  - Policy network architecture
  - RL formulation
  - Training Hyperparameters
- **New state-of-the-art** on three benchmarks
Graph HyperNetworks for Neural Architecture Search

Chris J. Zhang\(^1\), Mengye Ren\(^1,3\), Raquel Urtasun\(^1,3\)
\(^1\) Uber Advanced Technologies Group  \(^2\) University of Waterloo,  \(^3\) University of Toronto

Graph HyperNetworks

Motivation:
- Neural architecture search is an expensive nested optimization
  \[
  a^* = \arg \min_a L_{val}(w^*(a), a), \quad w^*(a) = \arg \min_w L_{train}(w, a)
  \]
- Instead of using SGD to learn weights, use trained hypernetwork to generate weights
- Graph HyperNetworks (GHN) explicitly model the topology of architectures by learning on a computation graph representation

Anytime Prediction

<table>
<thead>
<tr>
<th>Method</th>
<th>Search Cost (GPU days)</th>
<th>Param (\times 10^6)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMASHv1 (Brock et al., 2018)</td>
<td>?</td>
<td>4.6</td>
<td>94.5</td>
</tr>
<tr>
<td>SMASHv2 (Brock et al., 2018)</td>
<td>3</td>
<td>16.0</td>
<td>96.0</td>
</tr>
<tr>
<td>One-Shot Top (F=32) (Bender et al., 2018)</td>
<td>4</td>
<td>2.7 \pm 0.3</td>
<td>95.5 \pm 0.1</td>
</tr>
<tr>
<td>One-Shot Top (F=64) (Bender et al., 2018)</td>
<td>4</td>
<td>10.4 \pm 1.0</td>
<td>95.9 \pm 0.2</td>
</tr>
<tr>
<td>Random (F=32)</td>
<td>-</td>
<td>4.6 \pm 0.6</td>
<td>94.6 \pm 0.3</td>
</tr>
<tr>
<td>GHN Top (F=32)</td>
<td>0.42</td>
<td>5.1 \pm 0.6</td>
<td>95.7 \pm 0.1</td>
</tr>
<tr>
<td>NASNet-A (Zoph et al., 2018)</td>
<td>1800</td>
<td>3.3</td>
<td>97.35</td>
</tr>
<tr>
<td>ENAS Cell search (Pham et al., 2018)</td>
<td>0.45</td>
<td>4.6</td>
<td>97.11</td>
</tr>
<tr>
<td>DARTS (first order) (Liu et al., 2018b)</td>
<td>1.5</td>
<td>2.9</td>
<td>97.06</td>
</tr>
<tr>
<td>DARTS (second order) (Liu et al., 2018b)</td>
<td>4</td>
<td>3.4</td>
<td>97.17 \pm 0.06</td>
</tr>
<tr>
<td>GHN Top-Best, 1K (F=32)</td>
<td>0.84</td>
<td>5.7</td>
<td>97.16 \pm 0.07</td>
</tr>
</tbody>
</table>

ImageNet Mobile: Comparison with NAS methods which employ advanced search methods (e.g. RL)

<table>
<thead>
<tr>
<th>Method</th>
<th>Search Cost (GPU days)</th>
<th>Param (\times 10^6)</th>
<th>FLOPs (\times 10^6)</th>
<th>Accuracy</th>
<th>Top 1</th>
<th>Top 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>NASNet-A (Zoph et al., 2018)</td>
<td>1800</td>
<td>5.3</td>
<td>564</td>
<td>74.0</td>
<td>91.6</td>
<td></td>
</tr>
<tr>
<td>NASNet-C (Zoph et al., 2018)</td>
<td>1800</td>
<td>4.9</td>
<td>558</td>
<td>72.5</td>
<td>91.0</td>
<td></td>
</tr>
<tr>
<td>AmoebaNet-A (Real et al., 2018)</td>
<td>3150</td>
<td>5.1</td>
<td>555</td>
<td>74.5</td>
<td>92.0</td>
<td></td>
</tr>
<tr>
<td>AmoebaNet-C (Real et al., 2018)</td>
<td>3150</td>
<td>6.4</td>
<td>570</td>
<td>75.7</td>
<td>92.4</td>
<td></td>
</tr>
<tr>
<td>PNAS (Liu et al., 2018a)</td>
<td>225</td>
<td>5.1</td>
<td>588</td>
<td>74.2</td>
<td>91.9</td>
<td></td>
</tr>
<tr>
<td>DARTS (second order) (Liu et al., 2018b)</td>
<td>4</td>
<td>4.9</td>
<td>595</td>
<td>73.1</td>
<td>91.0</td>
<td></td>
</tr>
<tr>
<td>GHN Top-Best, 1K</td>
<td>0.84</td>
<td>6.1</td>
<td>569</td>
<td>73.0</td>
<td>91.3</td>
<td></td>
</tr>
</tbody>
</table>
We learn a data-dependent latent generative representation of model parameters, and perform gradient-based meta-learning in this low dimensional latent space.

The resulting approach, Latent Embedding Optimization (LEO), decouples the gradient-based adaptation procedure from the underlying high-dimensional space of model parameters.

LEO is state-of-the-art on both miniImageNet and tieredImageNet 5-way 1-shot and 5-shot classification tasks.
Proximal Meta-Policy Optimization: ProMP
Jonas Rothfuss*, Dennis Lee*, Ignasi Clavera*, Tamim Asfour, and Pieter Abbeel

Goal
1. Analyze credit assignment in meta-reinforcement learning
2. Develop a new objective that trains for the pre-update sampling distribution

Credit Assignment Sampling Distribution

Low Variance Curvature Estimator (LVC)

\[ J^{\text{LVC}}(\tau) = \sum_{t=0}^{H-1} \pi_\theta(a_t | s_t) \left( \sum_{t'=t}^{H-1} r(s_{t'}, a_{t'}) \right) \quad \tau \sim P(\tau) \]

- Meta-gradient with low variance
- Unbiased closed to local optima

Proximal Meta-Policy Optimization: ProMP

ProMP Objective:

\[ J_T^{\text{ProMP}}(\theta) = J_T^{\text{CLIP}}(\theta') - \eta D_{KL}(\pi_{\theta_o}, \pi_{\theta}) \quad \text{s.t.} \quad \theta' = \theta + \alpha \nabla_{\theta} J_T^{LR}(\theta) \]

Experiments

Incorporates the benefits of:
- Proximal Policy Optimization
- LVC Estimator
Attentive Task-Agnostic Meta-Learning for Few-Shot Text Classification

Xiang Jiang\textsuperscript{1,2}, Mohammad Havaei\textsuperscript{1}, Gabriel Chartrand\textsuperscript{1}, Hassan Chouaib\textsuperscript{1}, Thomas Vincent\textsuperscript{1}, Andrew Jesson,\textsuperscript{1} Nicolas Chapados\textsuperscript{1}, Stan Matwin\textsuperscript{2} \\
\textsuperscript{1}Imagia Inc. \textsuperscript{2}Dalhousie University

**Task-agnostic** representation learning

**Task-specific** attentive adaptation

Attention **decouples** the representation learning
Variadic Meta-Learning by Bayesian Nonparametric Deep Embedding
Kelsey Allen, Hanul Shin*, Evan Shelhamer*, Josh Tenenbaum

few-shot learning small-scale \[\rightarrow\] supervised learning large-scale

**variadic meta-learning**

Any-shot, any-way generalization between meta-train and meta-test with mixed supervision

experiments:
- from 5-way to 1692-way and from 1-shot to unsupervised on Omniglot
- from 1-shot to 50-shot on mini-ImageNet
- from 2-shot to 5000-shot on CIFAR-10

with comparison of prototypes, MAML, graph nets, and good old supervised learning

---

**BANDE** clusters labeled and unlabeled data into multi-modal prototypes that represent each class by a set of clusters instead of only one

---

**multi-modal prototypes**

for alphabet and character recognition

<table>
<thead>
<tr>
<th>Training</th>
<th>Testing</th>
<th>Proto. Nets</th>
<th>BANDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphabet</td>
<td>Alphabet</td>
<td>64.9±0.2</td>
<td>91.2±0.1</td>
</tr>
<tr>
<td>Alphabet</td>
<td>Chars (20-way)</td>
<td>85.7±0.2</td>
<td>95.3±0.2</td>
</tr>
<tr>
<td>Chars</td>
<td>Chars (20-way)</td>
<td><strong>94.9±0.2</strong></td>
<td><strong>95.1±0.1</strong></td>
</tr>
</tbody>
</table>
From Nodes to Networks: Evolving Recurrent Neural Networks

Aditya Rawal*, Risto Miikkulainen*

aditya.rawal@uber.com, risto@cs.utexas.edu

* Work done at Sentient Technologies

Meta-LSTM: Seq2Seq model to predict learning curve. Speeds-up search by 4X.

LSTM

NAS Cell

Evolved Cell

Language Modeling

Music

Encourage Search for Novel Cells

Recurrent Cell as Tree
Defaults commonly used in Machine Learning research and practise
Meta Learning for Defaults – Symbolic Defaults

Jan N. van Rijn, Florian Pfisterer, Janek Thomas, Andreas Müller, Bernd Bischl, Joaquin Vanschoren

- Defaults commonly used in Machine Learning research and practise
- Example: SVM($C=1.0$, $\gamma=0.0125$, kernel=RBF)
Defaults commonly used in Machine Learning research and practise

Example: SVM\((C=1.0, \gamma=0.0125, \text{kernel=RBF})\)

Goal: Defaults based on meta-feature

Example: SVM\((C=85, \gamma=0.2 / \text{num. features, kernel=RBF})\)

Classical form of meta-learning

Question: How to find good symbolic defaults?

Answer: Let’s discuss this at our poster!